

Contenido

Sintaxis completa de la instrucción SELECT	1
Nuestro GRUPO POR.....	1
¿Qué expresiones podemos poner en SELECT?	2
Uso de funciones de agregado con grupos.....	.
Condiciones HAVING.....	4
Orden de ejecución de instrucción SELECT	5
EJERCICIOS BD JARDINERÍA.....	¡ ¡Error! Marcador no definido.

Sintaxis completa de la instrucción SELECT

Con GROUP BY la sintaxis completa instrucción **SELECT** queda de esta forma:

```
SELECCIONAR [DISTINCT]{ComaListaDeExpresiones[*]}  
FROM ORIGEN DE LOS DATOS  
  
[WHERE condicionesDeFila]  
  
[GROUP BY ComaListaCriterisAgrupació]  
  
[HAVING condicionesDeGrup]  
  
[ORDENAR POR ComaListaDeExpresiones];
```

Nuestro el GRUPO POR

En el apartado GROUP BY, se indica el nombre de las columnas o expresiones (cualquier expresión válida para la cláusula SELECT en una consulta por filas a excepción de * DISTINCT) por las que se agrupa. el

Se creará un grupo por cada valor diferente en los criterios especificados en el GROUP BY (incluido el NULL). Toda fila perteneciente al origen de datos (resultado del FROM) que cumpla la condición del WHERE, pertenecerá a uno de los grupos que se forman.

Puede verse como que todas las filas que 'salen' del FROM y que pasan los filtros del WHERE (conjunto principal) se dividen en subconjuntos disjuntos por lo que toda fila pertenece a uno de ellos.

Es decir, aunque una fila tuviera valor NULL en los criterios del GROUP BY, se crearía el grupo de valor NULL en los criterios y ese grupo (o subconjunto) podría tener una o más filas.

Como es obvio, todas las filas pertenecientes a un mismo grupo tienen los mismos valores en los criterios del GROUP BY. Por tanto, los valores de los criterios del GROUP BY son una característica del grupo y, como tales, se pueden proyectar en el SELECT.

¿Qué expresiones podemos poner en SELECT?

Hay que diferenciar muy bien las consultas sobre filas (nos devuelven una proyección por cada una de las filas seleccionadas) de las consultas con GROUP BY que nos devolverán una proyección (fila) por cada uno de los grupos.

Si estamos haciendo una consulta de grupos, sólo podemos proyectar en el SELECT características de grupo de acuerdo a los criterios de agrupación.

Lo vemos con un ejemplo:

En una consulta sobre los alumnos del SERPIS en la que agrupamos por curso, no podemos proyectar la fecha de nacimiento o el nombre porque estos no son características de grupo, son características individuales de cada alumno.

Sí que podría preguntarse, por ejemplo: ¿Cuántos alumnos tiene cada grupo? ¿Cuál es la nota media de cada grupo? ¿Cuál es la nota máxima o mínima que se ha dado en cada grupo?...También podría preguntarse ¿Cuál es el grupo?

En una consulta con GROUP BY, en SELECT podemos proyectar FUNCIONES DE AGREGADO, CRITERIOS DEL GROUP BY y/o FUNCIONES SOBRE LOS CRITERIOS DEL GROUP BY (esto último es menos frecuente).

Por ejemplo: sea la tabla EXISTENCIAS

Tipo	Modelo	Almacén	Cantidad
Arkansas	6	1	2500
Arkansas	6	2	5600
Arkansas	6	3	2430
Arkansas	9	1	250
Arkansas	9	2	4000
Arkansas	9	3	678
Arkansas	15	1	5667
Arkansas	20	3	43
COLOMBIA	10	2	340
COLOMBIA	10	3	23
COLOMBIA	38	1	1100
COLOMBIA	38	2	540
COLOMBIA	38	3	700

Si ahora ejecutamos la siguiente instrucción:

```
SELECT Tipo,Model
DE Existencias
AGROUPAR POR Tipo,Modelo;
```

Si agrupamos en base a las columnas Tipo y Modelo se crearán 6 grupos ya que hay 6 valores diferentes para los criterios. Obtendremos este resultado:

GRUPO AR 6

Tipo Modelo Almacén Cantidad			
Arkansas	6	1	2500
Arkansas	6	2	5600
Arkansas	6	3	2430

GRUPO AR 20

Tipo Modelo Almacén Cantidad			
Arkansas	20	3	43

GRUPO AR 9

Tipo Modelo Almacén Cantidad			
Arkansas	9	1	250
Arkansas	9	2	4000
Arkansas	9	3	678

GRUPO BI 10

Tipo Modelo Almacén Cantidad			
CON UN	10	2	340
CON UN	10	3	23

GRUPO AR 15

Tipo Modelo Almacén Cantidad			
Arkansas	15	1	5667

GRUPO BI 38

Tipo Modelo Almacén Cantidad			
CON UN	38	1	1100
CON UN	38	2	540
CON UN	38	3	700

Por cada Tipo y Modelo diferente se ha creado un grupo. Una consulta de grupo devuelve una fila por grupo que puede tener las columnas (proyecciones)que consideramos.

Los datos Almacén y Cantidad no están disponibles ya que varían entre las filas del grupo.

Es decir, esta consulta es errónea:

```
SELECT tipo, modelo, cantidad
FROM existencias
AGROUPAR POR tipo,modelo;
```

JUSTO

La razón es el uso de la columna cantidad, en el SELECT. Como no se ha agrupado por ella, se podría producir un error en la mayoría de GBD.

En el caso de MySQL no da error; si proyectamos campos o expresiones que no forman parte de los criterios del GROUP BY aunque el valor que proyecta en la mayoría de los casos no tendrá sentido: será un 'al azar' de los valores de alguna de las filas del grupo. En otros GBD, hacerlo podría producir un error.

En una consulta de agrupación (GROUP BY) sólo se pueden proyectar los criterios de agrupación (son comunes en todas las filas del grupo) y funciones de agregado. Esto es lo que sería sintácticamente correcto.

Uso de funciones de agregado con grupos

Lo normal es agruparlo para obtener cálculos sobre cada grupo. Por ejemplo, podemos modificar la consulta anterior de esta forma:

```
SELECT tipo, modelo, SUM(Cantidad)
FROM existencias
AGROUPAR POR tipo, modelo;
```

Así sabríamos la cantidad total que se tiene de un determinado tipo y modelo en todos los almacenes. Obtendríamos este resultado:

Tipo	Modelo	SUM(Cantidad)
Arkansas	6	10530
Arkansas	9	4928
Arkansas	15	5667
Arkansas	20	43
CON UN	10	363
CON UN	38	1740

Se suman las cantidades para cada uno de los grupos. Igualmente podríamos realizar cualquier otro tipo de cálculo (AVG, COUNT,...)

Condiciones HAVING

Al igual que la cláusula WHERE restringe las filas seleccionadas, la cláusula HAVING restringe los grupos seleccionados.

HAVING recogerá la condición(es) que deben cumplir los grupos.

En HAVING se pueden poner condiciones que afectan a funciones de agregado o a criterios del GROUP BY.

Las funciones de agregado no se pueden restringir en WHERE. La razón reside en el orden en el que se ejecutan las cláusulas de la instrucción SELECT. Se calcula primero el WHERE y después los grupos (cláusula GROUP BY); por lo que no podemos usar en el WHERE la función de cálculo de grupos SUM, porque es imposible en ese momento conocer el resultado de la SUMA al no haberse establecido todavía los grupos.

La solución es utilizar otra cláusula: **HAVING**, que se ejecuta una vez realizados los grupos. Es decir, si queremos ejecutar condiciones sobre las funciones resumen, debe realizarse en la cláusula **HAVING**.

Por ejemplo, si sólo nos interesan los grupos con una cantidad total de más de 500 unidades, la consulta anterior quedaría:

```
SELECT tipo,modelo, cantidad, SUM(Cantidad)
FROM existencias
AGROUPAR POR tipo,modelo
HAVING SUM(Cantidad)>500;
```

Esto no implica que no se pueda utilizar WHERE. Esta expresión sí es válida:

```
SELECT tipo,modelo, cantidad, SUM(Cantidad)
FROM existencias
DONDE tipus!='AR'
AGROUPAR POR tipo,modelo
HAVING SUM(Cantidad)>500;
```

Por otra parte, al ser el campo tipo un criterio del GROUP BY es una característica de cada grupo y también podríamos haber restringido su valor en HAVING. La siguiente consulta daría un resultado análogo.

```
SELECT tipo,modelo, cantidad, SUM(Cantidad)
FROM existencias
AGROUPAR POR tipo,modelo
HAVING SUM(Cantidad)>500 AND tipo!='AR';
```

Orden de ejecución de instrucción SELECT

Para evitar problemas conviene conocer la forma de trabajar de una instrucción SELECT.

Estos son los pasos en la ejecución de una instrucción de agrupación por parte del gestor de bases de datos:

[1] Seleccionar las filas deseadas utilizando WHERE .

[2] Se establecen los grupos indicados en la cláusula **GROUP BY**

[3] Se calculan los valores de las funciones totales (**COUNT, SUM, AVG,...**) .

[4] Se filtran los registros (resultados de los grupos) que cumplen la cláusula **HAVING**

[5] El resultado se ordena sobre la base del apartado **ORDER BY**.